A Hybrid Artificial Neural Network Gravitational Search Algorithm for Rainfall Runoffs Modeling and Simulation in Hydrology

[1]*A.A. Ojugo,* [2]*Emudianughe, J.,* [3]*R.E Yoro,* [4]*E.O. Okonta and* [5]*A.O. Eboka*
[1,2]*Mathematics/Computer Science and Earth Science Departments, Federal University of Petroleum Resources Effurun, Delta State. ojugo_arnold@yahoo.com, emusjul@yahoo.co.uk*
[3]*Department of Computer Science, Delta State Polytechnic Ogwashi-Uku, Delta State, rumerisky@yahoo.com*
[4,5]*Department of Computer Science Edu., Federal College of Education-Technical, Asaba, Delta State, okeyokonta@yahoo.com, an_drey2k@yahoo.com*

Abstract

Artificial Neural Network (ANN) as a method of data processing and inspired by studies of the nervous systems – has become a robust tool for modeling complex, non-linear and dynamic processes due to its flexible mathematical structure that easily generalize patterns with results even with imprecise, noisy and ambiguous input data. This work describes ANN's application to implement a model to simulate runoff at the Benin Owena River Basin Developmental Agency (BORDA) – with data collected from four (4) gauge and six (6) stream-flow stations namely: Benin, Ekpoma, Sapele and Agbor catchments respectively. The study uses the 4G SE-design; the structured analysis of the existing is based on the lumped, conceptual hybrid (HBV and TOPMODEL) used for calibration and validation. The existing system's bottleneck includes large computational demand, excessive parameter requirement with validation, still an on-going process. Its mean annual rainfall of Benin, Ekpoma, Sapele and Agbor stations are 823, 732, 962 and 734mm respectively with computed COE of 58, 24, 56 and 42% respectively – indicating strong inter-annual and spatial variability in sub-catchments. Variation in the annual rainfall was observed and long-term runoff trend reflects, the effect of variation cycle with significant correlations between rainfall and runoff as observed in sub-catchments (via historic dataset obtained for the period). The study contributes as: (a) soft computing (a branch of Artificial Intelligence) with an aim to create a synergy with other fields/disciplines, and in this case (hydrology) in its bid to implement the hybrid ANNGSA algorithm for RR process. It also contributes in Artificial Intelligence (AI) – as it aims to create machine/system that mimics the human brain – so that such systems (in this case – hybrid ANNGSA model) that will train the ANN network to simulate future flood occurrence, provide lead time warning for flood management.

***Keywords*:** *Catchment, Algorithms, Evolutionary, Fitness Function*

1. Introduction

Soft Computing (SC) as a field aims to merge Artificial Intelligence with other fields of endeavors to create a synergy and new field, dedicated to solve problems by exploiting numeric data and human knowledge simultaneously through the use of mathematical models and symbolic reasoning – to yield a technique, tolerant to imprecision, uncertainty, partial truth and noise in its input data via optimization. Thus, such models end up as soft as the human brain [1,2].

Real world optimization requires tuning to be robust, so that even with noise employed at its input, it yields an output solution that is guaranteed of high quality. Research in a bid to tune search methods have helped to advance the field of Evolutionary Algorithms – capable of performing both quantitative (numeric) and qualitative data processing that ensures qualitative statements of knowledge and experience in form of natural languages [3]. SC components spans across several branches as inspired by evolution, natural laws and behavioural patterns in biological populations. These includes Genetic Algorithm, Artificial Neural Network, amongst other – all of which are meta-heuristic optimization for constraint satisfaction problems in vector space made up of intelligent agents that searches a space for its optimal fitness. Thus, SC mimics natural agents seeking food and have proven efficient in complex optimization [4]. [5,6] notes that ***robust*** optimization has three feats in its attempts to explore dynamic processes namely:

Progress in Intelligent Computing and Applications
Volume2, Number1,March 2013
doi: 10.4156/pica.vol2.issue1.2

a. ***Robustness*** helps to estimate system's effectiveness even with noise implementation.
b. **Continuous adaptation** (yields agents void of local minima, introduces random immigrants of high diversity to slow convergence in the search space as well as balances data exploitation and exploration so that in learning the properties of change, it yields an accordingly biased solution).
c. ***Flexibility*** – decisions made with uncertainty has its impacts in a system's future state. Thus, optimization aims to predicts the future needs with an algorithm that focuses on both its objective function, to make the system ***flexible*** and facilitate adaptation (if necessary) with the ease of blackbox integration.

Environmental change occurs quickly. Thus, long-term projection prone to errors, adaptation expensive and technically impossible [7]. The study of neural networks derives from trials in an attempt to translate into mathematical models, principles of biological processing and generate, in the fastest time period, implicit and predictive model evolution of a system. In particular, NN derives from experience its ability to be able to recognize feats and behaviours from historic data so as to be able to "suggest" to the model, the optimal fitness of high quality and void of ***over-fitting***, irrespective of modification via other approximations that uses multiple agents. These cannot be ignored as they constantly affect quality of any solution [8]. This work illustrates adopts neural network model (combined with GSA to speed up the last stages of ANN) in RR modeling. Thus, we explore the structural differences and implications of multi-agent and multi-population models (as agents do not follow predetermined rules, but tend to create their own behavioral rules based on a model for rainfall-runoff of hydrological data).

The need arose from: (a) the dynamic nature of RR and its conceptual models that are flawed and filled with unfounded results, (b) some ANN use hill-climbing method whose solution may get stuck at local minima, (c) ANN's speed shrinks as its approaches global optima.

The study implements a hybrid (ANNGSA) algorithm for RR model, compares result with existing benchmarks via data generated from BORDA in Nigeria. GSA will help to speed up NN final stages and find robust optima in a shorter amount of time, in large and complex tasks.

2. Soft Computing and Optimization

Optimization deals with searching for optimal solution(s) in a given problem, chosen from set of possible solutions (***search***) space. The study considers input and output constraints – with uncontrollable parameters modeled in the ANN's hidden layer. Thus, are not explicitly present with a search space confined to real parameters often limited by lower/upper bound values. Soft Computing on the otherhand, deals with biologically and evolution inspired mathematical models that spans across various fields. Examples includes as thus:

2.1. Artificial Neural Network (ANN)

The structure of ANN as a data processing model is inspired by biological neurons – consisting of interconnected neurons or nodes (used as processing elements). Its major feat is its ability to ***learn*** by example via simulation, making them universal ***approximators***. The brain learns in its behavior to process data – and each neurons sends/receives electrochemical signal from others neurons via a host of fine structures called ***Dendrites***. These receives signals are re-sent to its axon so that the ***Synapse*** converts axon's activities and thus, learning can occurs by adjusting weight of the synapse [9]. Synapses are connecting links, characterized by weights whose input is summed by an ***adder***. The operations may vary depending on the task (linear combiner) at hand and its activation function to limit the NN output's amplitude. In a simple mathematical model, the synapses effect represented by weight connections help modulate the effects of associated inputs and the nonlinear feats exhibited by its nodes via ***transfer or activation*** function. Neuron's output is computed as weighted sum of input via a transfer function [10]. ***Learning*** is achieved by adjusting its weights via an algorithm, which defines a neuron's output in terms of its induced local field and neuron output is given by:

$$\emptyset = f(net) = f\sum_{i=1}^{m} Xi * Wij \qquad (1)$$

Encoded, ANN has three basic layers namely: input, hidden and output – and based on two configurations: ***feed-forward*** (signal flows from input to output without feedback and data processing extended over multiple layers of units); and ***recurrent*** (same as feddforward but with feedback, dynamic properties and an activation values that undergoes relaxation to evolve the network to a stable state where its activation values change no more. In some tasks, its output value change is significant such that the dynamic behavior constitutes network's output). These architectures is dependent on application area, feats and system requirement [11]. ANN is configured by applying a set of inputs produces a set of desired outputs. Various methods are used to set the connection strengths namely: (a) ***explicitly*** via apriori knowledge, and (b) train network via teaching it patterns that changes its weight based on a learning rule. Learning is divided into: ***supervised***, ***unsupervised*** and ***reinforcement*** [12].

[13-17] notes that in ***supervised*** learning, an input vector with a set of desired responses, one for each node, is relayed to the output. A forward pass is done and errors between ***desired*** and ***actual*** response for each node in the output is found, and then used to determine weight changes in the net based on the learning algorithm Thus, desired signals on output is provided by an external teacher. Example is back-propagation, delta rule and perceptron rule. In ***unsupervised*** learning (or self organization), its output unit is trained to respond to clusters of pattern at its input so that the system discovers statistically, salient features of the input population. It also has no prior knowledge into which patterns are classified; Rather, the system develops its own representation of input stimuli; while in ***Reinforcement*** is learning what to do, mapping situations to actions to help maximize a numerical reward signal. The learner is not told actions to take, as in most forms of machine learning, but instead must discover which actions yield the most reward by trying them. In some cases, the actions may affect not only the immediate reward, but also the next situation and, through that, all subsequent rewards. These two feats, trial/error search and delayed reward are its two distinguishing properties.

2.2. Gravitation Search Algorithm (GSA)

GSA is based on Newton's laws of gravity and motion with its main idea, being to consider isolated system of masses, where every mass represents a solution to a certain problem. Law of gravity states that every particle attracts another and the gravitational force between particles are directly proportional to the product of their masses and inversely proportional to distance between them [18]. So, an agent's performance depends on its mass as they attract each other via gravitational force (a pull towards those of heavier masses). Each N agents is initialized as thus:

$$Xi = (x_i^1 + x_i^2 + \ldots + x_i^d + x_i^n \qquad (2)$$

n is dimension of the problem, and also the position of the *i*th agent in the *d*th dimension. At start point of the solution, agents are situated randomly. At specific time, a gravitational force is defined as thus:

$$Fij = G(t) = \frac{Mi(t) * Mj(t)}{Rij(t) + \varepsilon} \{Xj(t) - Xi(t)\} \qquad (3)$$

M_i and M_j are objects (i and j) masses, $R_{ij}(t)$ is Euclidean distance between the two, G(t) is gravitation constant at time t and ε is a small constant. The randomly initialized gravitational constant *G*, decreases by time t to control the search's accuracy. Thus *G* is a function of initial value (*G0*) and time (*t*). Total force acting on agent *i* in the dimension *d* is thus:

$$F_i^d = \sum_{j \in kbest, j \neq 1} rand(i) * Fij \qquad (4)$$

rand – randomizes agents'initial states between intervals [0,1]. The acceleration of agent i, at time t, in *dth* dimension is directly proportional to force acting on that agent, and inversely proportional to agent's mass by:

$$Aid(t) = \frac{Fid(t)}{Mij(t)} \qquad (5)$$

The next velocity of an agent is a function of its current velocity plus its current acceleration. Next position and velocity of an agent is calculated as thus:

$$Vid(t+1) = rand(i) * Vid(t) + Aid(t) \qquad (6)$$
$$Xid(t+1) = Xid * Vid(t+1) \qquad (7)$$

$V_i^d(t)$ is agent velocity in *d*th dimension at time t, and rand is a random number between [0,1]. Mass is calculated via fitness evaluation and are updated as:

$$Mi(t) = \frac{Fit(i) - worst(t)}{best(t) - worst(t)} \qquad (8)$$

Fit(t) is fitness value of an agent i at time t. Best(t) and worst(t) indicates the strongest and weakest agents based on to their fitness route. For a Max task, they are defined:

$$worst(t) = \max_{j \in \{1,2 \ldots N\}} Fit(t) \qquad (9)$$

$$best(t) = \min_{j \in \{1,2 \ldots N\}} Fit(t) \qquad (10)$$

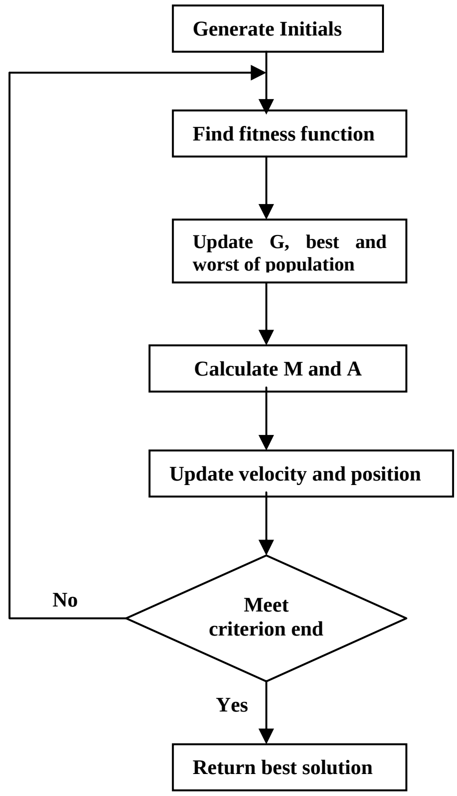

Figure 1. Steps for gravitational search algorithm

At start, agents are located as solution points in the search space such that with each cycle, the positions and velocities of agents are updated via Eq. (5, 6 and 7). G and M as calculated are updated with each iteration or move, and stopped when an optimal solution is found. GSA use exploration (ability to navigate the space) and exploitation (ability to find optima around a good solution) in the shortest time. Exploration steps guarantee the choice of values or parameters of the random agents; while exploitation steps allows agents of heavier masses to move more slowly in order to attract those of lesser mass [19].

3. Methods and Materials

The study area selected is the Benin-Owena River basin of Nigeria with landmass of 22045km^2, mean rainfall of 846mm annually and perennial discharge of 3.8m^3/s (dry periods) and 15m^3/s (peaks). The area's elevation ranges at various sub-stations ranging from 816 and 2178m a.s.l. Average slope is 48° and most slopes have south-eastern orientation. Soil texture is mostly loam and clay due to the swampy nature. Forests dominated by shrubs and timber covers approximately 64% of the catchment area with its statistical parameter as in table 1.

Table 1. Statistical Parameter for Rainfall Runoff for Benin Catchment Area (2003 – 2008)

Area	Mean	Std Dev	Coeff. Of Variance	Max Rainfall	Min Rainfall
Benin	823	359	58	4532	142
Ekpoma	732	299	24	1034	102
Sapele	962	420	56	4320	127
Agbor	734	343	42	1354	156

The historic dataset collected at the Benin catchment is from 2003 – 2008; and for this study – the dataset is split into three: ***training*** (45%), ***cross-validation*** (25%) and ***validation*** (30%) – since there is no hard rules in data splitting. All three fragment starts at period of constant low discharge and rainfall.

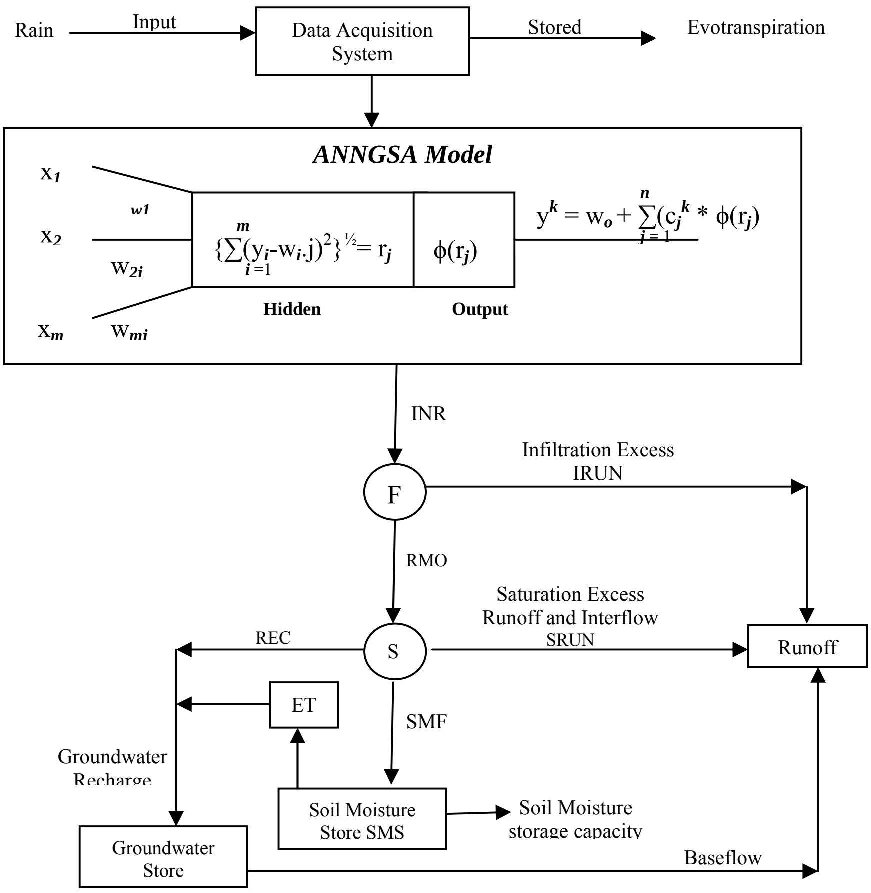

Figure 2. Block diagram of the envisaged ANNGSA model design

3.1. Experimental Models

The study adopts ***unsupervised*** learning employed on the TLRN architecture with 3 datasets; while RBF is used as a control model to compare the results. The TLRN has input weights, transfer function to control output and learning laws to define the importance of input weights. The goal is to process data by training them to generate satisfactory results and provide a fail-safe to eradicate noise in a data-stream acquired via Online-Data Acquisition System (DAS) so that data is processed at real-time. The network learns from experiences, generalized from previous datasets to new ones with abstract feats, at its inputs containing irrelevant data. The model's control parameters are neuron's ***weights*** and its ***biases;*** and an output layer with a neuron called ***runoff***. The ***radial*** basis function is implemented as ***control*** model to check results from our MLP-design [21].

Model's nonlinear states are identified at its input and output data selection during training and testing, model structure selection, parameter estimation and validation. Thus, the training data must represent watershed feats and meteorological patterns in runoff modeling. Input variable (rainfall) are selected to describe the physical phenomena of the process. Thus, we employ a 5year time-series data of ***Benin*** catchment to evaluate the model's performance with the data split into three sets: ***training***, ***cross-validation*** and ***testing***. The study aims to illustrate model's ability in simulating future runoff occurrences, without including land-use properties of watersheds.

Trial-error is used in selecting number of hidden layers and nodes in each hidden layer. Thus, NN with one hidden layer can be used – for to increase the number of parameters by adding more hidden layers, complicates ***training*** – though the network is complex enough to accurately simulate dynamic and/or nonlinear feats [22]. Standard tasks use 15, 30, 45, 60 and 100 hidden nodes (on each layer) to examine model's performance and our study however, adopts ***single*** hidden layer with 18-hidden nodes. Thus, we adopt a total of 18-input signals (with regards to evapo-transpiration, rainfall and previous discharge at the four stations). Previous studies and preliminary ***results*** indicate that NN with a hidden layer outperform those of two or more. The optimal hidden layer size is found by systematically increasing the number of hidden node until network's performance shows no further improvement or it longer improves significantly.

3.2. Model Performance Evaluation

The model's performance is evaluated via its computed coefficient of efficiency (COE), mean square error (MSE), mean absolute error (MAE) and mean relative error (MRE) – as these are most commonly used performance measures in hydrological modeling. MSE, MRE and MAE will have an ideal value of 0; while COE will aim to show the model's efficiency with an ideal value of 1 as seen in equations (8), (9) and (10). Y_{pi} and Y_{oi} are ***predicted*** and ***observed*** output values, n is observations over which errors are computed. A model with minimum error is considered, best choice.

$$MSE = {}^{1}\!/_{n} \sum_{i=1}^{m} \{(Ypi - Yio)2\}^{1/2} \qquad (10)$$

$$MAE = {}^{1}\!/_{n} \sum_{i=1}^{m} |Ypi - Yio| \qquad (11)$$

$$MRE = {}^{1}\!/_{n} \sum_{i=1}^{m} \frac{|Ypi - Yio|}{Yio} \qquad (12)$$

Model validation/testing here, is an undertaking that is not and should not be carried out by a single researcher or research group; but, requires a scientific dialogue. Improper model applications and its ambiguously presented results sometimes impede such dialogue. The aim of such pitfalls, is to a great extent minimize as well as to reduce confusion in hydrological modeling.

4. Result Presentation Tradeoffs

Hydrological simulation is a feat not to be undertaken by any one method. Thus, tradeoffs in the result of each one researcher and/or research groups will fall under the following:

a. **Result Presentation** – Researchers with their flaws, often prefer to modify or build new models rather than re-test limitations, biasness and inabilities of existing ones – since negative results are less valuable [23]. Knowledge models are good examples, where limitations are not clearly stated, as they are often overrated [24,25]. Thus, modelers use many methods and misleading graphs to compare simulation or observations as justified [26,27]. This study's data driven models employs evolutionary stochastic method to cub non-linearity and dynamism with historic datasets, used to train, cross-validate and test such unlike with knowledge driven models.

b. **Model Efficiency –** [28] defined model's efficiency as analogous to coefficient of determination – R^2, now widely termed goodness-of-fit. Figures used to show how well simulations is in agreement with observations often provide limited data as long runoff series are squeezed, and lines for observed and simulated runoff are not easily distinguishable. Some authors do not provide numerical data; but rather states that the model is in 'good agreement' with observations [29,30]. Even when a measure of goodness is given, it does not always provide the relevant information.

c. **Insufficient Model Testing –** The most used model validation is simulation, compared with observed runoff for a period not used in test. Studies seldom exist where the result of such test is 'unsuccessful' [31] and authors cannot publish negative results. An exception is [32] where poor validation results are reported. So many authors validate poor results [33,34]. [35] predicted the response of runoff to climatic changes without any kind of validation of the calibrated models. Many studies suffer from inadequate data, the distributed model seldom demonstrates superiority over lumped. If a model aims to simulate more than runoff, such ability is demonstrated [27]. Development of complex models based on limited data gives misleading results [36]. A ***second*** issue is drawing unfounded conclusions in testing. [37] conclude that the close agreement between analytical and numerical results underscores the utility of Muskingum-Cunge routing as a viable and accurate method for routine applications in flood hydrology.

4.1. Model Performance Analysis

Tables 2-5 show comparative performance values between the TLRN and RBF models. Results as in Table 1 (Benin, 18 inputs); Table 2 (Sapele, 18 inputs); Table 3 (Ekpoma, 17 input nodes) and Table 4 (Agbor, 17 input nodes) – implies that the performance of the model is improved during testing with a greater level of efficiency of the catchment (i.e. proper network training and criteria selection). Furthermore, COE for Benin substation is better than the Sapele substation (it is probably due to the size of the substation that contribute to the neural modeling). COE, MSE, MRE, and MAE reflect that the RBF consistently outperforms the TLRN (MLP) – and the RBF model can be trained much faster than MLP (though NN performance is hardly influenced by level of non-linearity and training data selection). Number of neurons in hidden layer significantly influences network's performance. If number is small, network may not achieve its accuracy – and too many nodes result in overtraining. Fully developed sub-areas (Benin and Sapele) generate higher peak flood discharge during training and generalization. Use of two hidden layers is a merit in larges substation (Benin and Sapele); while for smaller catchments – it is sufficiently handled by single hidden layer NN model structure. Obviously, the application of neural network method in modeling the relationship between rainfall and runoff for these catchment areas is quite appropriate.

Table 2. Simulated Values from Benin Station

Model	***Training Phase***				
	I-H-O Structure	COE R	MSE cumecs	MAE cumecs	MRE cumecs
TLRN	18-18-1	0.635	0.926	0.665	1.208
RBF	18-input	0.982	0.982	0.618	0.969
Cross Validation Phase					
TLRN	18-18-1	0.723	0.886	0.712	1.109
RBF	18-input	0.892	0.889	0.567	0.901
Testing Phase					
TLRN	18-18-1	0.641	0.654	0.518	1.385
RBF	18-input	0.966	0.596	0.442	1.510

Table 3. Simulated Values from Sapele Station

Model	I-H-O Structure	COE R	MSE cumecs	MAE cumec	MRE cumecs
	Testing Phase				
TLRN	18-18-1	0.723	0.910	0.710	1.328
RBF	18-input	0.832	0.945	0.623	0.789
	Cross Validation Phase				
TLRN	18-18-1	0.713	0.902	0.712	1.021
RBF	18-input	0.821	0.891	0.70	0.901
	Testing Phase				
TLRN	18-18-1	0.714	0.723	0.628	1.108
RBF	18-input	0.833	0.756	0.512	1.310

Table 4: Results at the Ekpoma Station

Model	I-H-O Structure	COE R	MSE cumecs	MAE cumecs	MRE cumec
	Training Phase				
MLP	17-17-1	0.552	0.920	0.532	1.109
RBF	17-input nodes	0.812	0.956	0.621	0.961
	Cross Validation Phase				
MLP	17-17-1	0.823	0.621	0.629	0.915
RBF	17-input nodes	0.827	0.684	0.721	0.891
	Testing Phase				
MLP	17-17-1	0.641	0.654	0.518	1.113
RBF	17-input nodes	0.966	0.596	0.442	1.065

Table 5: Results at the Agbor Station

Model	I-H-O Structure	COE R	MSE cumecs	MAE cumec	MRE cumecs
	Training Phase				
MLP	17-17-1	0.632	0.800	0.620	1.280
RBF	17-input	0.732	0.882	0.610	0.789
	Cross Validation Phase				
MLP	17-17-1	0.621	0.822	0.702	0.867
RBF	17-input	0.761	0.811	0.691	0.921
	Testing Phase				
MLP	17-17-1	0.714	0.723	0.628	1.180
RBF	17-input	0.833	0.756	0.512	1.109

4.2. Discussion and Findings

The network models used for this study are the TLRN and RBF with one hidden layer – a hybrid (NNGSA) adopted from [20] – and the results reflect that the model's performance is satisfactory and feasible for RR-model in the Benin catchment of Benin-Owena Basin. The study noted that a ***model's*** inaccuracy is clarified by longer period of training data (so long overtraining does not occur) with many peak discharge as the work was carried out using the 5years period historic dataset of the rainfall-runoff records in four (4) areas of BORDA, in the Niger Delta Region of South-South Nigeria.

Though the general pattern of rainfall at the stations was similar over the period, annual totals indicates both spatio-temporal variability in rainfall over Benin and Sapele catchment. There was also inter-annual and inter-monthly variability in the climate of Benin-Owena over the past century. Runoff from all stations display a similar general pattern to rainfall which both shows an increase that had a peak in 2010. Gauging stations in Ekpoma and Agbor had lesser runoffs because rainfall increases from Southwest to Southeast in Benin catchment. The lack of similarity in the runoff pattern from the same substations indicates high spatial variability in runoff. The gauging stations show differences in runoff – indicates differences in the average rainfall received, land use type or soil type amongst other hydrological feats. There is today, significant relationship between rainfall and runoff.

[37] notes that the cohesive relationship exist between rainfall and runoff due to a considerable temporal and spatial variability exhibited by RR-process (various physical mechanisms that governs the process's dynamics). A major rainfall feat of rainfall within semi-arid and swampy areas of Nigeria, is that it comes as rainfalls and convective thunderstorms that are highly isolated resulting in a high spatial variability. Factors that affect runoff are more uniform for smaller catchments and it is expected that their coefficients of determination will increase with decrease in area. However, this was not the case as smaller catchment had almost similar results and sometimes higher coefficients of determination, which demonstrates that rainfall variance is same as runoff variance since variability of actual evo-transpiration is small relative to variability of annual precipitation and runoff.

4.3. Benchmark Comparison

Results from [38] compared against those obtained from this study notes that various attempts to improve ***conceptual*** runoff models have often resulted in frustrating conclusions. Our study notes the various result tradeoffs in the model evaluation, and though model can be modified towards a better description of the real processes, the quality of runoff simulations does not increase significantly. For the HBV model as adopted by [38], such experience has been reported for tests of different formulations for various models, a measure of evaporation and snowmelt [39], insertion of evaporation depending with altitude [40] and the use of an explicit interception routine [41]. Good results in terms of runoff simulations are obtained with different and even unrealistic concepts. The difference in numerical measures of *Reff* is often small, even in cases where the model performance increased after modifications [42].

Unsatisfactory results in [38] are partly attributed to the nature of the model's evaluation as it is not the case in this study. It was noted that a model's efficiency assesses its goodness, which is not often sensitive to runoff improvements during low flow conditions – as such improvements may vanish in a simulated runoff. Thus, progress made in use of remote data-sensing helps to provide new ways to parameterize and validate models [43]. Land-use classification for modeling is often derived from satellite data and spatial distribution of some variables can be estimated via extension of surface saturated areas, soil moisture in areas with no or only little vegetation. Remotely-sensed quantities may be used as proxies to other variables [44] and these will further increase the total information extractable from the two types of data. The study notes that difficulties in the use of remotely-sensed data are in its limited availability and costs. The increased computing power may be utilized in different ways – that allows refining the resolution of distributed models or include additional process representations so as to enlarge a model's complexity via executing more calculations per model run.

The use of data-driven models via SC may be used to address model uncertainty of Monte-Carlo procedures employed in conceptual/knowledge models [45]. The study noted that quantification of prediction uncertainties is of central importance, especially in practical applications, given the large uncertainties associated with the use of runoff models. For HBV model, [46] proposed use of quantification for runoff forecasts in operational use. Model complexity must aim at improves the model's testability, often a limited value with routines that cannot be tested against the data.

5. Conclusion and Recommendations

The study adopted TLRN and RBF, whose goal is to process data via previously trained (and cross-validated) and tested algorithm using historic dataset so as to generate satisfactory results. The network learns from experience by generalizing from previous examples to new ones, and abstract characteristics from inputs that may contain irrelevant data. The main control parameters are its weights and biases; and its output layer has one neuron ***runoff***. SC via AI methods are successfully used to model complex/dynamic relationships and studies indicates, NN have proven to be useful tools in hydrological modeling and flood management.

The model observed that the rainfall and runoff variability of the Benin catchment is considered both temporarily and spatially, and its implications on surface water resources should be explored in flood management. Data from 4 gauge-stations in the catchment (22045km^2) is used to (a) develop GSANN model to simulate RR, and (b) determine performance via known benchmark. The mean annual rainfall of the four rain gauge stations Benin, Ekpoma, Sapele and Agbor are 823, 732, 962 and 734mm respectively, with COE of 58, 24, 56 and 42% respectively – strong inter-annual and spatial variability in sub-catchments. Variation in annual rainfall was observed and long-term trends in runoff, reflects the effect of cycle variations with significant correlations between rainfall and runoff as observed in sub-catchments (via historic dataset obtained for the period 2006 – 2010).

Models are not more fiction than a representation of reality as they may provide good and useful fiction – and the primary value of models may be their use as an intellectual tool, to help us better understand and reflect on reality. By this, models support experts to make estimates about the future, but models alone cannot provide these estimates and simulations. As thus, the model is recommended for use in RR-model simulation. Thus, these recommendations were made from the study:

1. Parameter uncertainty is a significant source of uncertainty in model predictions. Thus, model predictions can and should be inputted in ranges, computed via Integral Monte-Carlo methods, rather than as single values.
2. Multi-criteria training with adequate datasets can help to reduce parameter uncertainty.
3. Simulations and model prediction have limited practical use without clear data about its reliability and accuracy.
4. This data-driven model has only been calibrated against runoff and it may not provide reliable simulations cum prediction of internal variables like groundwater levels.
5. Validation via differential split-sample testing will acts a powerful tool and is essential for the further development of a model for two reasons: (a) identification of weak parts and (b) evaluation of improvements.

6. References

[1] Abarghouei, A.A., Ghanizadeh, A and Shamsuddin, S.M., "*Advances in soft computing methods in edge detection*, J. Advance Soft Comp. Applications, ISSN: 2074-8523, 1(2), 2009.

[2] Abbott, M., Bathurst, J., Cunge, J., O'Connell, P and Rasmussen, J., "*Introduction to Syst`eme Hydrologique Europ´een, "SHE", 1 and 2: History and philosophy of a physically-based, distributed modeling system"*, J. of Hydrol., 87, pp 45–77, 1986.

[3] Shi, Y., "*Particle swarm optimization"*. IEEE Transaction on Neural Networks, pp 8-13, 2004.

[4] Sedighizadeh, D and Masehian, E., "*Particle Swarm Optimization: Taxonomy, Methods and Applications"*, J. of Computer Theory and Engineering, ISSN: 1793-8201, Vol. 1(5), 2009.

[5] Abraham, A., "*Handbook of Measuring System Design"*, John Wiley and Sons Ltd, ISBN: 0-470-02143-8, pg 901 – 918, 2005.

[6] Beven, K. J., "*How far can we go in distributed hydrological modeling?"*, Hydrol. Earth Sys. Sci., 5, 1–12, SRef-ID: 1607-7938/hess/2001-5-1, 2001.

[7] Beven, K. J., "*Rainfall-Runoff Modeling: the Primer"*, JohnWiley & Sons, Chichester, pg 102-243, 2001.

[8] Ursem, R.K., Krink, T., Jensen, M.T and Michalewicz, Z., "*Analysis and modeling of control*

tasks in dynamic systems". IEEE Transactions on Evo. Comp. 6(4), Pg 378-389, 2002.

[9] Caudill M. (1987): *Neural Networks Primer, Part I*, AI Expert December: pg 46-52.

[10] Coello, C. A., Pulido, G. T and Lechuga, M.S., (2004): *Handling multiple objectives with particle swarm optimization*, In congress on Evol. Computation, Vol. 8, pg 256–279

[11] Bishop, C. M., (1995). *Neural networks for pattern recognition*. Oxford University Press. pg 253-294.

[12] Openshaw, W.R., (2003): *Rainfall runoff processes: A workbook to accompany online module*, [online]: http://www.engineering.usu.edu/dtarb/rrp.html.

[13] Conway, A. J., Macpherson, K. P., and Brown, J. C., (1998): *Delayed time series predictions with neural networks*, Neurocomputing Vol. 18, pg 81–89.

[14] Dawson, C and Wilby, R., (2001): *Comparison of neural networks in river flow forecasting*, Hydrol. Earth Sci., SRef-ID: 1607-7938/hess/2001-3-529.

[15] Dawson, C and Wilby, R: (2001): *Hydrology modeling using neural networks*, Prog. Phys. Geog., 25, pg 80–108.

[16] Guo, W.W., (2010), "A novel application of neural networks for instant iron-ore grade estimation," *Expert Systems with Applications*, vol. 37, pp. 8729–8735.

[17] Imam, T., (2012). "Intelligent computing and foreign exchange rate prediction: What we know and we don't," *Progress in Intelligent Computing and Applications*, vol. 1, no. 1, pp. 1-15.

[18] Rashedi, E., Nezamabadi-pour, H and Saryazdi, S., (2009). *GSA: A Gravitational Search Algorithm*, Information Sciences, Vol. 179(13), (2009), pp. 2232-2248.

[19] De Vos, N. J., Rientjes, T. H and Pfister, L., (2005): *Groundwater levels as indicator in rainfall-runoff modeling using Artificial Neural Networks*, Proceedings NCR 2005.

[20] Dibike Y. B. and Solomatine D. P. (1999). "*River Flow Forecasting Using MLP Neural Network*" J. of Physics and Chemistry of Earth: EGS1.1-0002.

[21] French, M. N., Krajewski, W. F., and Cuykendall, R. R., (1992): *Rainfall forecasting in space and time using a neural network*, J. Hydrol., 137, 1–31.

[22] Gaume, E and Gosset, R., (2003): *Over-parameterization, obstacles of artificial neural networks in hydrology?*, Hydrol. Earth Sys. Sci., 7, 693–706, SRef-ID: 1607-7938/hess/2003-7-693.

[23] Govindaraju R. S. (2000): *Artificial Neural Networks In Hydrology*, *Part II: Hydrologic Applications*", J. of Hydrology and Engineering, 5(2): 124-135.

[24] Gupta, H. V., Sorooshian, S and Yapo, P. O., (1998): *Improved calibration of hydrologic model: Multiple and non-commensurable measures of data*, Water Resources, Vol. 34(4), pg 751–763.

[25] Hall, M. J., (2001): *How well does your model fit the data?*, J. Hydroinformatics, 3(1), 49–55.

[26] Varoonchotikul, P., (2003): *Flood forecasting using artificial neural networks*, Ph.D. thesis, Swets & Zeitlinger, Lisse, The Netherlands.

[27] Halff, A. H., Halff, H. M., and Azmoodeh, M., (1993): *Predicting from rainfall using neural networks*, ASCE Proceedings of Engineering Hydrology, pg 760–765.

[28] Haykin, S., (1999): *Neural networks, a comprehensive foundation*, New Jersey: Prentice Hall.

[29] Nash, J. E. and Sutcliffe, J. V. (1970): *River flow forecasting through conceptual models; part I – a discussion of principles*, J. Hydrol., 10, 282–290.

[30] Heppner, H and Grenander, U (1990): *A stochastic non-linear model for coordinated bird flocks*, In Krasner, S (Ed.), *The ubiquity of chaos* (pp. 233–238). Washington: AAAS.

[31] Hjemfelt, A. T. and Wang, M., (1993): *Artificial neural networks as unit hydrograph applications*, ASCE Proceedings of Engineering Hydrology, 754–759.

[32] Hsu, K. L., Gupta, H. V., and Sorooshian, S., (1995): *Artificial neural network modeling of the rainfall-runoff process*,J. Water Res., 31 (10), 2517–2530.

[33] Jain, A. and Srinivasulu, S., (2004): *Development of effective and efficient rainfall-runoff models using integration of deterministic, real-coded GA and ANN*, Water Res., 40(4), W04302.

[34] Jang, J.S., (1993): *Adaptive fuzzy inference systems*. IEEE Trans. Syst., Man Cyber., 23, 665–685.

[35] Karunanithi, N., Grenney,W. J., Whitley, D and Bovee, K. (1994): *Neural network for river flow prediction*, J. Comput. Civil Eng., 8 (2), 201–220.

[36] Khondker, M.U.H., Wilson, G and Klinting, A., (1998): *Application of neural networks in real time flash flood forecasting*. Proc. of 3rd Conf. on Hydroinformatics, 777–781. Copenhagen: Balkema, Rotterdam.

[37] Kitanidis, P. K. and Bras, R. L., (1980): *Real-time forecasting with a conceptual hydrologic model, 2, applications and results*, Wat. Resour. Res., 16 (6), 1034–1044.

[38] Tarboton, D.G., (2003): *Rainfall runoff processes: A workbook to accompany online module*, [online]: http://www.engineering.usu.edu/dtarb/rrp.html, last retrieved: 16-8-2012.

[39] Lindstrom, G., Johansson, B., Persson, M., Gardelin, M and Bergstrom, S. (1997): *Test and development of the distributed HBV-96 hydrological model*, J. Hydrol., 201, 272–288.

[40] Reggiani, P. and Rientjes, T., (2005): *Flux parameterization in the representative elementary watershed approach: Application to basin*, Water Resour. Res., 41(4), W04013.

[41] Rientjes, T., (2004): *Inverse modelling of the rainfall-runoff relation; a multi objective model calibration approach*, Ph.D thesis, Delft University of Technology, Delft, Netherlands.

[42] Minns, A. W., (1998): *Artificial neural networks as sub-symbolic process descriptors*, Ph.D. Thesis, A. Balkema, Rotterdam, The Netherlands.

[43] Nishimura, S and Kojiri, T., (1996). *Real-time rainfall prediction using neural network and genetic algorithm with weather radar data*, 10th. Congress of the Asia and Pacific Division of the International Association for Hydraulic Research, 204-211.

[44] Rajurkar, M. P., Kothyari, U. C and Chaube, U. C., (2004): *Modeling of the daily rainfall-runoff relationship with artificial neural network*, J. Hydrol., 285, 96–113.

[45] Sajikumar, N. and Thandaveswara, B. S., (1999): *A non-linear rainfall runoff model using an artificial neural network*, J. Hydrol., 216, 32–55.

[46] Seibert, J., (2000): *Multi-criteria calibration of a conceptual runoff model,* Hydrol. Earth Sys. Sci., 4, 215–224, SRef-ID: 1607-7938/hess/2000-4-215.